Utrecht University

Faculty of Science
Ylias Ben Salah - 8936021
July 12, 2021

A comparison between Bayesian penalized regression priors: lasso and regularized horseshoe

Abstract

A comparison is performed between Bayesian penalized regression priors: the lasso and regularized horseshoe using the statistical programming language R. This study aims to provide researchers with insights into the use of these priors to deal with high-dimensional data. Therefore, the shrinkage behavior of the Bayesian lasso and regularized horseshoe models, using different hyperparameter settings, were compared. Furthermore, variable selection was executed for the models. Lastly, the predictive performances were evaluated based on their Root Mean Square Error (RMSE). Results showed that researchers have to take several factors into consideration. First consideration concerns which prior is best suited on their data. The Bayesian lasso showed more variation in shrinkage behavior and is easy to implement, while regularized horseshoe prior is more robust to their specific hyperparameter settings and is complex to implement. Second, researchers should consider a variable selection method. This paper shows that an RMSE plot is a suitable tool for variable selection. In conclusion, there were no significant differences in predictive performances found between the Bayesian lasso and regularized horseshoe.

Master Thesis Applied Data Science

Thesis commission:
Dr. Sara van Erp (supervisor)
Dr. Joran Lokkerbol (co-reader)

Table of contents

Introduction

In the current technological era, companies and researchers have more access to data than ever before. Technical devices such as phones, smartwatches, and computers are able to track user's behaviour and collect all kinds of data within a country's privacy laws. These huge amounts of data, also known as *Big Data*, offers great research possibilities in different fields. For example, a study showed how the American healthcare system can be improved using high-dimensional data from several sources (Bates, Saria, Ohno-Machado, Shah, & Escobar, 2014). Besides the research possibilities of high-dimensional data, there are also challenges concerning overfitting models when using high-dimensional data. Hence, more research can be done on certain models to deal with overfitting to provide better performing models for big data analysis.

Proven statistical methods such as linear regression models are less usable for these high-dimensional data due to computational limitations (Hawkins, 2004). Linear regression models are popular for evaluating the relative impact of a predictor variable on a particular outcome, but they perform poorly on datasets with large number of coefficients. Overfitting occurs when the model includes more coefficients than necessary, resulting in a model with a high variance hurting the generalizability (McNeish, 2015). The effect of overfitting increases when more complex models are tried to fit.

Penalized regression offers a solution for fitting models to high-dimensional data. While ordinary least squares (OLS) regression minimizes the sum of squared residuals (SSR) to find the estimates for regression coefficients, penalized regression also includes a penalty term to the minimalization of the SSR. This is in order to shrink small coefficients towards zero, while leaving the large coefficients. The shrinkage of non-relevant variables towards zero makes this technique popular for datasets with a large number of predictors. Penalized regression techniques avoid overfitting and achieve model parsimony (Derksen & Keselman, 1992; Tibshiranit, 1996). A widely researched and implemented penalization method is the least absolute shrinkage prior, also known as *lasso*. Lasso makes it possible to shrink coefficients towards zero with the possibility of setting some coefficients to exactly zero, resulting in a simultaneous estimation and variable selection procedure (McNeish, 2015).

Penalized regression can also be used in the Bayesian framework. Penalized regression in the Bayesian framework has several advantages over classical penalization and offers multiple shrinkage priors. These shrinkage priors shrink variables towards zero which can guide variable selection. Small coefficients are more likely to be shrunken towards zero, causing them to be excluded from the model. Priors come in different shape and form, such as the Bayesian lasso and the more flexible regularized horseshoe. The Bayesian lasso does not distinguish between the coefficients and thus shrinks them all by the same amount. This prior is easy to implement but could result in too much shrinkage of relevant coefficients (Park & Casella, 2008). In addition, the regularized horseshoe does distinguish between the coefficients due to its flexibility. However, the amount of shrinkage depends on multiple hyperparameters which makes it a complex prior to implement (Piironen & Vehtari, 2017).

This paper aims to provide researchers with insight into the use of the Bayesian lasso and regularized horseshoe with different model settings. While the lasso is available in both the classical and Bayesian frameworks, the complex regularized horseshoe is only available in the Bayesian framework. Interesting is to understand to what extent the shrinkage differs among the priors. Therefore, the Bayesian lasso will be compared with the Bayesian regularized horseshoe using data analysis. This paper could be used as guidance for researchers when using regularized horseshoe or Bayesian lasso for model fitting.

To begin with, the methodology is given wherein the penalized regression method and the Bayesian priors are explained. Furthermore, the data wrangling and data analysis steps are also included in the methodology. In addition, the results of the data analysis are presented and interpreted. Finally, the results are discussed, and the conclusion is given.

Methodology

Ordinary least squares (OLS) regression

The standard OLS regression is a frequently used regular regression method to predict a metric outcome from predictors. This regression estimates the coefficients by minimizing the sum of squared residuals, wherein the residuals are equal to the difference between the observed and the predicted value (McNeish, 2015). A simple OLS regression can be presented as follows:

$$y = X\beta + \varepsilon, \tag{1}$$

Here, y is the n *x* 1 vector of outcome observations, X is the n *x* p + 1 matrix of the predictor variable including a vector of ones for the intercept, β is the parameter of the regression, and ϵ is the generally normally distributed n *x* 1 vector of errors. Herein, n stands for the sample size and p stands for the number of predictor variables. An issue with the OLS regression is that random noise can become entangled with signal. This occurs more often with small sample sizes relative to the number of predictors. This can result in overfitting, with the consequences of underestimated standard errors, and non-parsimonious models. Estimates from an overfit model perform well on the fitted sample sizes, but they perform significantly less on a different sample size of the same population, thus the generalizability of the model is in dispute (McNeish, 2015).

To create a model where no overfitting occurs for small sample sizes, the bias-variance trade-off of the regression model should be balanced properly. Herein, bias is the difference between the average prediction of the model and the correct value, and variance is the variability of a model prediction for a specific value. An overfitting model has a low bias and a high variance, which can be seen in the figure below:

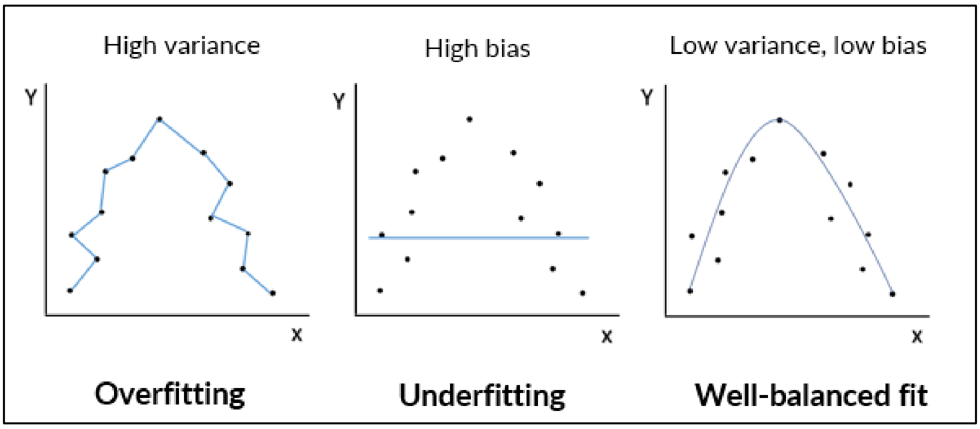

Figure 1 shows the bias-variance trade-off for overfitting, underfitting and balanced models.

Penalized regression

Penalized regression analysis is frequently used to prevent overfitting in small data samples with many predictors. The general idea of the penalization regression is to select relevant variables for a particular prediction using a statistical approach. While the OLS regression minimizes the sum of squared residuals, penalized regression adds a penalty term to the sum of squared residuals. The objective of the penalty term of the penalized regression is to shrink small coefficients towards zero while keeping large coefficients large. By doing so, a model can be created wherein no overfitting occurs and model parsimony is achieved, which results in model generalizability (Kyung, Gilly, Ghoshz, & Casellax, 2010). The formula of a general penalized regression is as followed:

$$\underset{\beta_0, \beta}{minimize}\left\{\frac{1}{2n}\|y - \beta_0 1 - X\beta\|\frac{2}{2} + \lambda_c\|\beta\|_q\right\},$$
$$where \ \|\beta\|_q = \left(\sum_{j=1}^{p} |\beta j|^q\right)^{\frac{1}{q}}. \qquad (2)$$

In equation 2, y is the vector containing the observations on the outcome variable with n-dimensions (y_1,,y_n). β_0 is the intercept, 1 represents the n-dimensional vector of ones, *X* represents a matrix of the observed scores on the predictor variables (n *x* p) and β is the parameter vector of regression coefficients with p-dimensions (β_1,…, β $_n$). λ_c represents the penalty parameter of the penalized regression. A higher λ_c value results in more shrinkage towards zero while $\lambda_c = 0$ results in ordinary least squares solutions. In addition, q in the equation determines the induced penalty type. There are several penalty types, each corresponds with the value of q, wherein q = 1 refers to the lasso penalty (Tibshiranit, 1996). Despite the promising classical lasso penalty technique, this study only focuses on the penalized regression method in a Bayesian framework.

Bayesian penalized regression

The Bayesian approach to penalized regression models requires priors for all model parameters and has multiple advantages over the classical framework. Only specific priors that provide clear shrinkage behaviour result in a Bayesian penalized approach. The form of the prior distribution decides the shrinkage behavior of the Bayesian penalized regression method. The more peaked the distribution is, the more shrinkage towards zero for small coefficients is executed. In addition, the heaviness of the tails in the distribution also affects the shrinkage behavior. Heavy tails allow large coefficients to escape the shrinkage towards zero.

Advantages of the Bayesian approach

The first advantage refers to the natural fit of penalization in a Bayesian framework. In any case, prior distributions are needed and parametric shrinkage towards zero can be accomplished by choosing a specific parametric form for the prior. Prior distributions can be chosen in order to shrink small effects (non-relevant parameters) to zero, and simultaneously keeping significant effects (relevant parameters) large. Furthermore, prior distributions combined with specific posterior estimate can result in the same outcome as classical penalization models and in some cases Bayesian priors perform even better than classical penalized methods (Kyung et al., 2010; Li & Liny, 2010). The second advantage lies in the estimation of the penalty parameter λ. Bayesian penalization makes it possible to estimate the penalty parameter with other model parameters in a single step. Instead of estimating the λ using cross-validation, Bayesian penalization sets a prior on the λ and estimates it using data. Large values for the λ results in heavier shrinkage towards zero and a λ value of zero results in no shrinkage (van Erp, 2020). The third advantage concerns the flexibility of the Bayesian penalization in terms of the penalty types that can be considered. While classical penalization methods result in a point estimate by finding the minimum of the penalized regression function, the Bayesian penalization technique uses Markov Chain Monte Carlo (MCMC) sampling which results in a full posterior distribution (van Erp, 2020).

Bayesian lasso

Interpreting lasso estimates as a Bayes posterior using Laplace distribution model was first noted by Tibshirani (1996). Park and Casella (2008) considered a fully Bayesian analysis by using MCMC sampling for the lasso with the Laplace prior distribution and extending their model by placing prior distributions on the σ^2 and λ to take care of the hyperparameters uncertainty. The following formula of the Laplace prior was considered:

$$\pi(\beta|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_j|/\sqrt{\sigma^2}}. \qquad (3)$$

Herein, β is the regression parameter, σ^2 is the variance and λ is the penalty parameter. Conditioning on σ^2 is important, because it guarantees a unimodal full posterior (Park & Casella, 2008). The Bayesian lasso is a global prior. Thus, the exact form of the Bayesian lasso and its shrinkage behavior fully depends on the adjustable λ, which operates as a global shrinkage parameter. In comparison with the classical lasso, the Bayesian lasso does not shrink small coefficients to exactly zero. Hence, additional criteria are essential for selecting relevant variables when using the Bayesian Lasso.

Advantage of the Bayesian lasso is that it is easy to implement. Disadvantage of the Bayesian lasso is that this prior can lead to over shrinkage of large coefficients due its global shrinkage property (Polson & Scott, 2010). A solution to this problem is found in the Bayesian regularized horseshoe prior. This prior allows simultaneously global shrinkage on all coefficients and local shrinkage on large coefficients to prevent too much shrinkage.

Regularized horseshoe

The Bayesian horseshoe is a prior which adds a local shrinkage parameter into the priors' equation next to the global shrinkage parameter. The horseshoe prior is characterized by an asymptote at zero and heavy tails, which allow heavy shrinkage of small coefficients towards zero while leaving the large coefficients out of the shrinkage (Carvalho et al., 2010; Polson & Scott, 2010). However, small shrinkage on large coefficients might be necessary, because the heavy tailed horseshoe can lead to an unstable MCMC sample when there are large but weakly identified coefficients in the data (van Erp, 2020). The regularized horseshoe offers solutions to the disadvantages of the horseshoe. Regularized horseshoe shrinks the small coefficients in the same way as the horseshoe prior, but it also guarantees small shrinkage on the large coefficients (Piironen & Vehtari, 2017). Regularized horseshoe is suggested in the following form:

$$\begin{aligned}
&\beta_j|\lambda_j,\tau,c \sim Normal\left(0,\tilde{\lambda}_j^2\tau\right),\ with\ \tilde{\lambda}_j^2 = \frac{c^2\lambda_j^2}{c^2+\tau^2\lambda_j^2},\\
&\lambda_j \sim half - Cauchy(0,1),\\
&\tau|\tau_0^2 \sim half - Cauchy(0,\tau_0^2), \quad with\,\tau_0 = \frac{p_0}{p - p_0}\frac{\sigma}{\sqrt{n}},\\
&c^2|v,s^2 \sim Inv - Gamma(v/2, vs^2/2).
\end{aligned} \tag{4}$$

In equation 4, the global parameter τ shrinks all β_j towards zero, the local shrinkage parameter λj allows certain β_j to escape the shrinkage, parameter c guarantees small shrinkage with *v* = 4 and $s^2 = 2$ and p_0 denotes a guess of the number of relevant variables . Modification of these shrinkage parameter lead to specific shrinkage behaviour, wherein *v* has to be small to ensure a robust pattern of shrinkage on the large coefficients (Piironen & Vehtari, 2017; van Erp, Oberski, & Mulder, 2019). The Bayesian regularized horseshoe is more flexible in comparison with the Bayesian lasso due its to the number of modifiable parameters.

Bayesian lasso tuning with brms

The Bayesian Regression Models using 'Stan' `(brms)` package is used in R for the data analysis. This package implements Bayesian multilevel models in R using the probabilistic programming language Stan, including several priors such as the Bayesian lasso and the regularized horseshoe. Furthermore, `brms` allow specifying multilevel generalized linear models with the same formula form as classical multilevel generalized linear models in the `glmer` function in R (Bürkner, 2017).

In `brms`, the Bayesian lasso has the following formula:

$$\beta \sim Laplace\left(0,\frac{scale\ *\ 1}{\lambda}\right). \tag{5}$$

In equation 5, the coefficient parameter β is given in a Laplace distribution with a mean of zero and a variance of scale *x* 1, divided by the inverse of λ. The scale and the inverse λ are modifiable hyperparameters which influence the prior distribution. The inverse λ value is given by the degrees of freedom (df) in a chi-square distribution, wherein a higher df value gives a less peaked distribution with thicker tails which results in less shrinkage. In addition, a lower scale gives a more peaked distribution with thin tails which result in heavier shrinkage towards zero. Figure 2 shows Bayesian lasso prior distributions with the default hyperparameters, with a modified scale, and with a modified df.

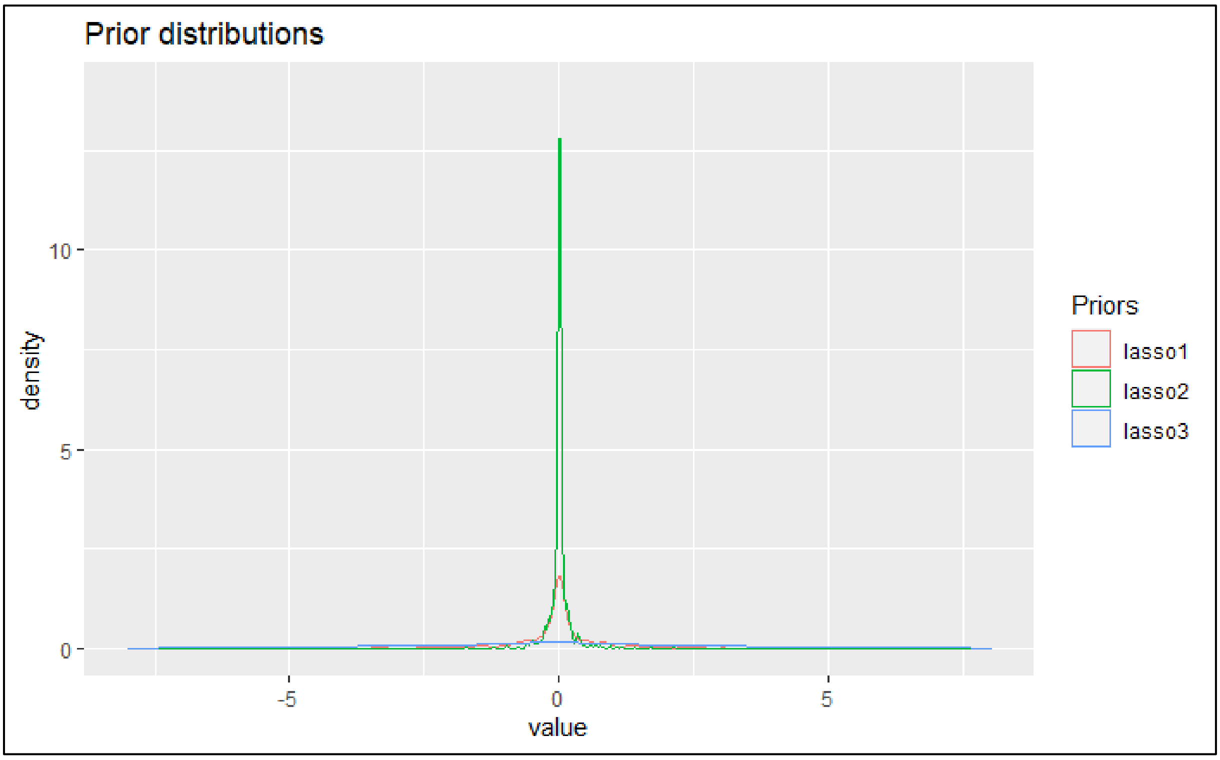

Figure 2 shows with default hyperparameters values [df = 1, scale = 1], lasso2 with hyperparameter values [df=1, scale=0.1], lasso3 with hyperparameter values [df = 5, scale = 1].

Figure 2 shows that there is a significant difference in the exact form of the distribution, thus a difference in the executed shrinkage. Most shrinkage is executed when the prior is very peaked, this is the case with lasso2. Least shrinkage is executed when the prior is least peaked which is the case for Lasso3. Lasso3 has a very low peak and is more spread in comparison to the other lassos. Furthermore, figure 2 shows that default lasso1 is less peaked than lasso2, but more peaked than lasso3. Hence, shrinkage of this prior is less in comparison to lasso2 and heavier in comparison to lasso3. For this research, the lasso1, lasso2, and lasso3 Bayesian priors will be implemented on data to compare the variable selection output.

Regularized horseshoe tuning with brms

In `brms`, the regularized horseshoe prior has five modifiable hyperparameters: df, scale_global, df_global, scale_slab and df_slab. For global shrinkage, the scale (scale_global) and the degrees of freedom (global_df) need to be specified. The global scale influences how peaked the prior is, wherein a smaller scale results in more overall shrinkage of the coefficients. The global degrees of freedom determine the tail behavior of the prior, wherein a higher value results in lighter tails. The hyperparameter that needs to be modified for the local shrinkage is the local_df which allows large coefficients to escape the shrinkage, wherein higher values result in lighter tails. In addition, there are two hyperparameters to protect large coefficients from heavy shrinkage. Therefore the scale (slab_scale) and the df (slab_df) need to be specified (van Erp, 2020). For this study, the `brms` default values of regularized horseshoe will be compared with two models wherein the global scale and the local scale are adjusted to a lower value.

Theoretically, lowering the global scale should result in heavy shrinkage towards zero for all coefficients, while lowering the local scale should result heavier shrinkage on the large coefficients in comparison with the default settings. Adjusting these hyperparameters gives us insight into the specific influence of these hyperparameters on the shrinkage behaviour and on the model variable selection.

Data wrangling

For this research, data for chronic kidney diseases research is used from an open-source database (UCI, 2015). The original dataset contained 400 observations, 24 predictive variables such as *age, blood pressure* and *sugar levels*, with *class* as the outcome variable which refers to whether or not the participant has a chronic kidney disease. Hence, logistic regressions are used for the models. Furthermore, the dataset included missing data in both predictive and outcome variables. The `MICE` package in R is used to impute values for the missing data. Data imputation with `MICE` is a preferred method for dealing with missing data because it minimizes the bias and deals with uncertainty (van Buuren & Groothuis-Oudshoorn, 2011). In addition, dummy variables were created for all nominal variables. In addition, numeric variables were standardized because shrinkage only affects coefficients equally if they are on the same scale (Bürkner, 2017). Finally, the dataset was randomly divided into a training set and test set for cross-validation, wherein 70% of the data is used as a training set and 30% is used as a test set.

Data analysis

Upon completion of the data wrangling, the Bayesian lasso and regularized models were fit on the training set. Table 1 gives an overview of the modified hyperparameters per model. Other hyperparameters of the regularized horseshoe which are excluded from table 1 were set on the default settings. Also, default priors were used on the model parameters.

Table 1
Overview of the modified hyperparameters for the Bayesian lasso and regularized horseshoe priors.

Lasso	**Degrees of freedom (df)**	**Scale**
Setting 1 (default)	1	1
Setting 2	1	0.1
Setting 3	5	1
Regularized horseshoe	**Global scale**	**Local scale**
Setting 1 (default)	1	1
Setting 2	0.1	1
Setting 3	1	0.1

The following step was to select the relevant variables of the fitted Bayesian penalized models, which is realized with the `projpred` package (Piironen, Paasiniemi, & Vehtari, 2018). The suggest_size function and Root-Mean-Square-Error (RMSE) plot are both used for variable selection and are available in `projpred`. RMSE measures the difference between the observed values and the predicted values, wherein an RMSE of zero indicates a perfect fit (Kuhn & Johnson, 2013). RMSE plot visualizes the influence of model sizes on the RMSE. To select variables based on the plot, the decrease of RMSE is examined per additional variable. Only variables that significantly decrease the RMSE should be included. The suggest_size function suggests the number of relevant variables based on a heuristic decision rule.

Finally, the performance of each model with the selected variables is evaluated using cross-validation. Models with their selected variables were fitted on the test set using the default `brms` model to predict the outcome variables. These predicted outcome variables are compared with the observed outcome variables which results in an RMSE value. The steps above are performed for all the Bayesian lasso and regularized horseshoe models.

Results

In this section, the results are presented and interpreted. Firstly, the amount of shrinkage for the Bayesian lasso and regularized horseshoe models are visualized and compared using different hyperparameter settings. Secondly, the variable selection per model is given and interpreted. Lastly, the models are evaluated based on their RMSE outcomes.

Amount of shrinkage

Bayesian Lasso

The amount of shrinkage is analyzed for small and large coefficients of the dataset. The coefficient *sugar* (su_1 in the dataset) is used as the small coefficient and describes the patients' blood sugar level, while the coefficient *red blood cells* (rbc_normal in the dataset) is used as the large coefficient and describes the patients' red blood cells level. The different hyperparameter settings of the Bayesian lasso models were given in table 1. The Bayesian lasso posterior mean estimates of the su_1 and the rbc_normal coefficients are presented in tables 2 and 3. Furthermore, the Bayesian lasso posterior distributions of the su_1 and the rbc_normal coefficients are visualized in figures 3 and 4.

Table 2

Posterior mean estimates of the small su_1 coefficient per lasso model

	Posterior mean estimates	**Std. error**
Su_1		
Lasso 1	0.18	4.04
Lasso 2	0.10	1.76
Lasso 3	0.25	6.32

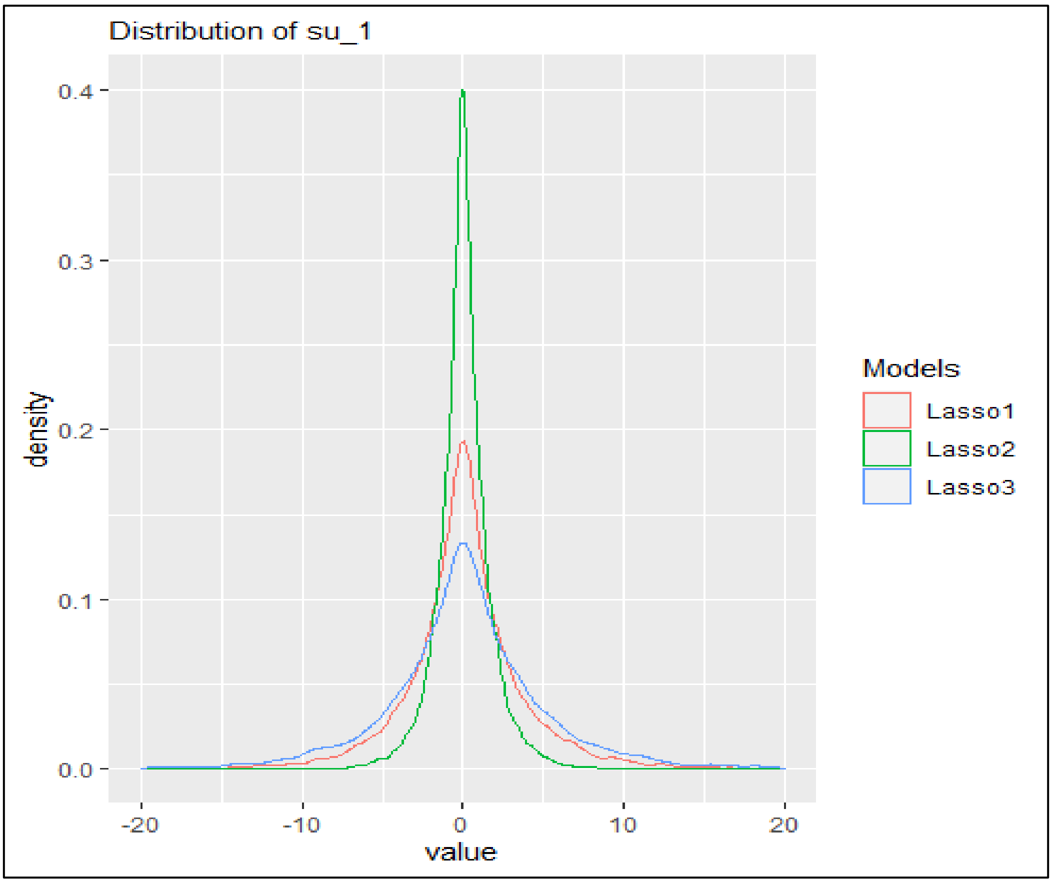

Figure 3 shows the posterior distribution of the estimated su_1 coefficient for three Bayesian lasso models.

Figure 3 shows that the posterior distributions of the three Bayesian lasso models lie around zero which implies that the su_1 coefficient is a non-relevant predictor. Posterior mean estimates of non-relevant coefficients should be close to zero, but this differs per model and depends on the shrinkage behavior of their priors. Figure 3 also shows that Lasso2 has the highest peak around zero with thin tails which indicates a high probability of su_1 having a posterior mean estimate of zero. Lasso1 has a lower peak and thicker tails, which indicates a higher probability of su_1 having a posterior mean estimate higher than zero in comparison to lasso2. Lasso3 has the lowest peak and thickest tails which indicates the highest probability of su_1 having a posterior mean estimate higher than zero in comparison to Lasso1 and Lasso2. In table 2, Lasso2 has the lowest posterior mean of su_1 with an estimated value of 0.18, while Lasso3 has the highest posterior mean of su_1 with an estimated value of 0.25. Thus, most shrinkage on su_1 is executed by Lasso2 and the least shrinkage is executed by Lasso3.

Table 3

Posterior mean estimates of the large rbc_normal coefficient per lasso model

	Posterior mean estimates	**Std. error**
Rbc_normal		
Lasso 1	-9.92	5.03
Lasso 2	-5.58	2.23
Lasso 3	-13.74	7.48

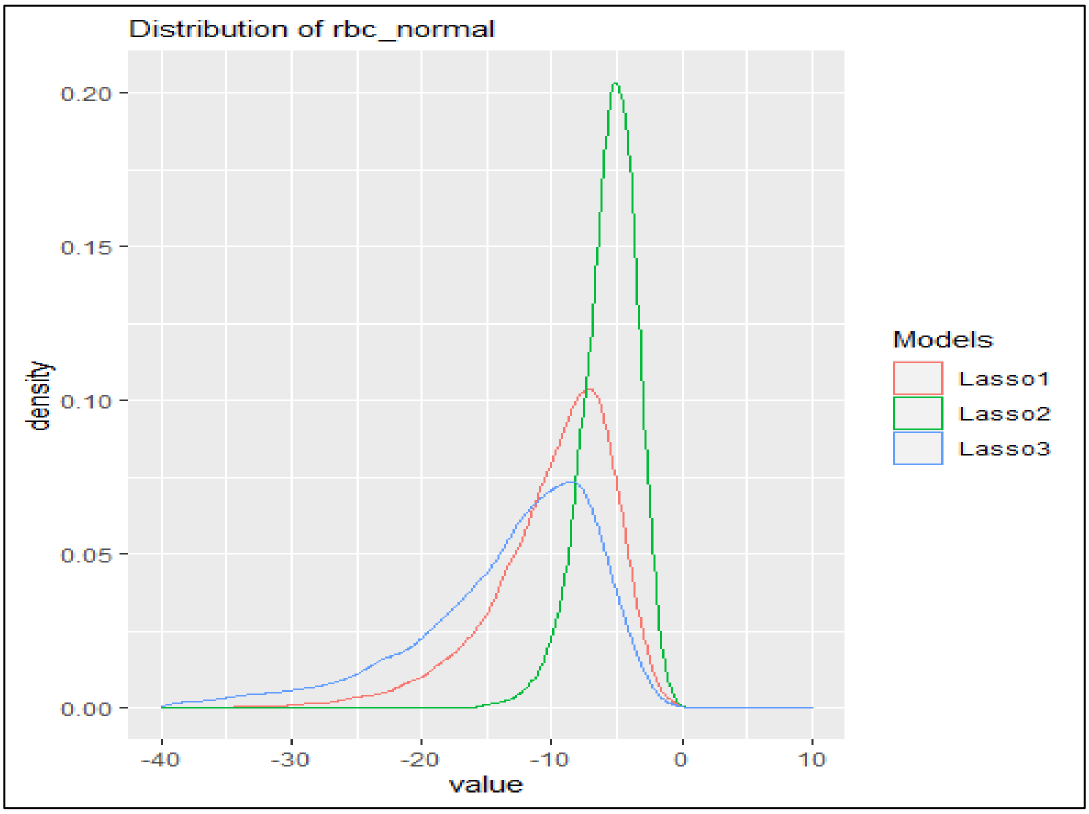

Figure 4 shows the posterior distribution of the estimated rbc_normal coefficient for three Bayesian lasso models.

In figure 4, the posterior distributions of the three Bayesian lasso models lie around a higher value than zero which indicates that rbc_normal is a relevant predictor. Furthermore, figure 4 shows that the three Bayesian lasso models do not centre around the same value, wherein a posterior distribution closer to zero indicates heavier shrinkage. The posterior distribution of Lasso2 is the closest to zero with the highest peak and thin tails which indicates a high probability for rbc_normal to have the lowest posterior mean estimate. Lasso1 has a lower peak and thicker tails which indicates a high probability of rbc_normal having a higher posterior mean estimate than lasso2. Lasso3 is the furthest to zero with the lowest peak and thickest tails which indicates the highest probability of rbc_normal having the highest posterior mean estimate in comparison to Lasso1 and Lasso2. Table 3 shows that lasso2 has the smallest posterior mean of rbc_normal with an estimated value of -5.58, while Lasso3 has the largest posterior mean of rbc_normal with an estimated value of -13.74. Thus, most shrinkage on rbc_normal is executed by Lasso2 and the least shrinkage is executed by Lasso3.

In figure 2 we saw that lasso 2 prior was most peaked with thin tails, while lasso 3 prior was less peaked with heavy tails. Based on these prior distributions, we would thus expect most shrinkage from lasso 2 and least shrinkage from lasso 3. This is in line with what we see here in figures 3 and 4.

Regularized horseshoe

The amount of shrinkage is also analyzed using regularized horseshoe models with different hyperparameter settings. These prior hyperparameter settings can be found in table 1. The regularized horseshoe posterior mean estimates of the su_1 and rbc_normal coefficients are presented in tables 4 and 5. In addition, the regularized horseshoe posterior distributions of su_1 and rbc_normal are visualized in figures 5 and 6.

Table 4

Posterior mean estimates of the small su_1 coefficient per regularized horseshoe model

	Posterior mean estimates	**Std. error**
Su_1		
Regularized horseshoe 1	0.09	2.60
Regularized horseshoe 2	0.12	2.52
Regularized horseshoe 3	0.10	2.20

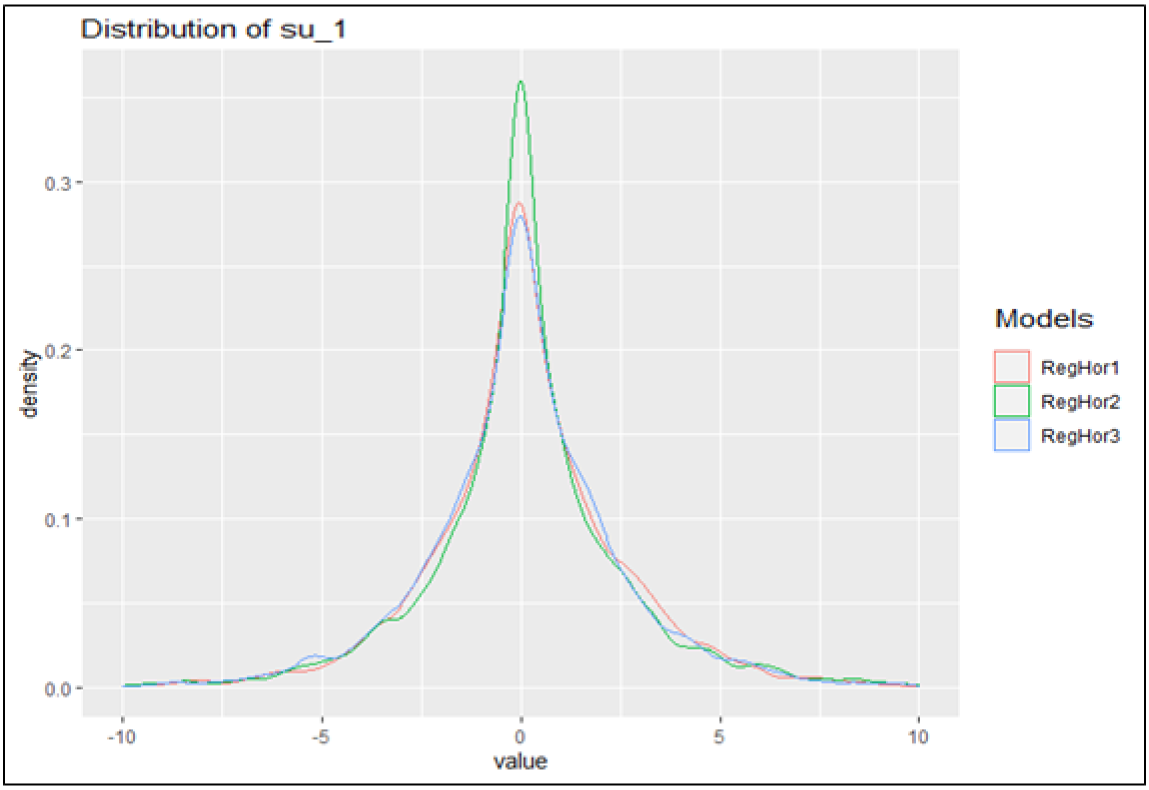

Figure 5 shows the posterior distribution of the estimated su_1 coefficient for three Bayesian regularized horseshoe models.

In figure 5 we see that all three regularized horseshoe models are centred around zero and have comparable posterior distribution. However, the posterior distribution of RegHor2 is more peaked than RegHor1 and RegHor3, but this difference is very small in comparison with the different lasso distribution peaks. Hence, the posterior distributions of figure 5 indicate a high probability of su_1 having a posterior mean estimate of zero for all three regularized horseshoe models. This is in line with table 4 which shows that the posterior mean estimates of the models are close to zero, wherein RegHor2 has the highest posterior mean estimate of 0.12 and RegHor1 has the lowest posterior mean estimate of 0.09.

Thus, the shrinkage behavior of the regularized horseshoe models on the small su_1 coefficient are almost identical. Even though RegHor2 has the highest peak, this does not translate into heavier shrinkage than the other two regularized horseshoe models. The slightly different posterior distribution peaks and posterior mean estimates are explainable by the number of MCMC samples. A lower number of MCMC samples result in less computation time but could also result in twisting outcomes.

Table 5

Posterior mean estimates of the large rbc_normal coefficient per regularized horseshoe model

	Posterior mean estimates	**Std. error**
Rbc_normal		
Regularized horseshoe 1	-6.66	2.92
Regularized horseshoe 2	-6.69	3.04
Regularized horseshoe 3	-5.49	2.63

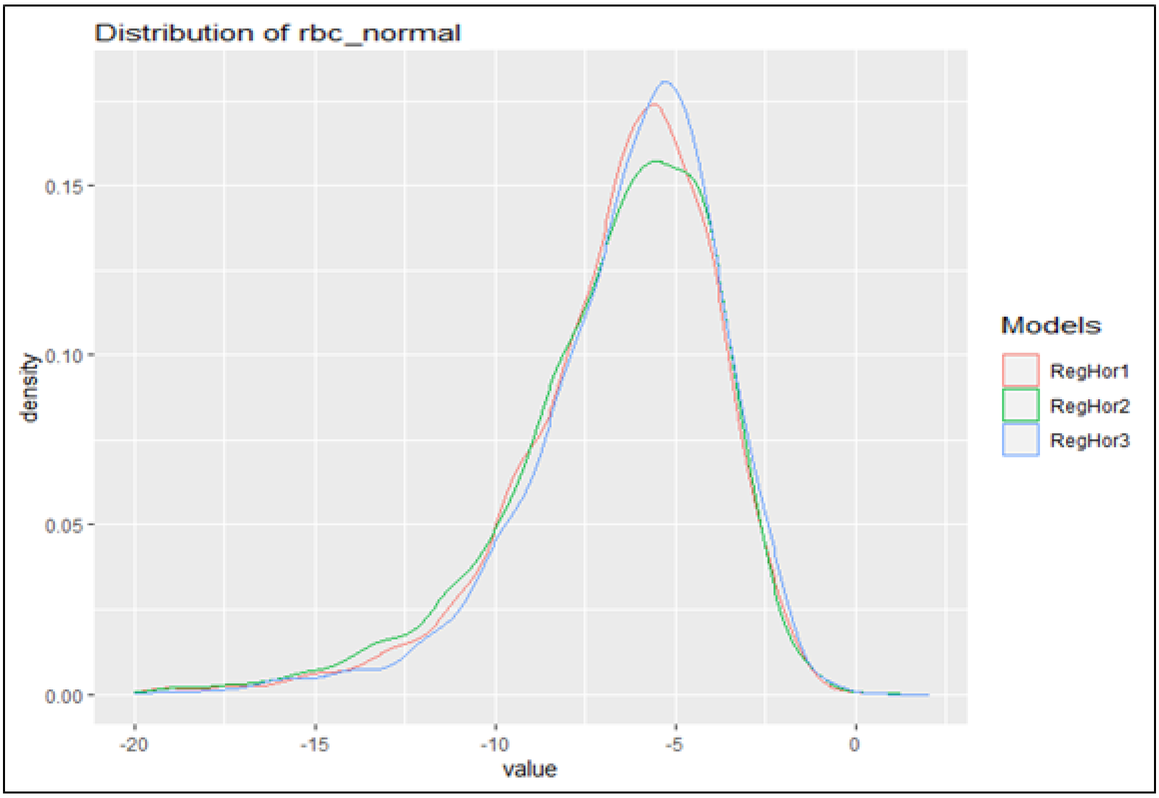

Figure 6 shows the posterior distribution of the estimated rbc_normal coefficient for three Bayesian regularized horseshoe models.

Figure 6 shows that all regularized horseshoe models are centred around the same value which is higher than zero. Furthermore, we see slightly different peaks of the posterior distributions with almost identical tails. Despite the fact RegHor2 has the lowest peak and slightly thicker tails than RegHor1 and RegHor3, this does not imply less shrinkage. Table 5 shows that RegHor3 has the lowest posterior mean estimate of -5.49 and RegHor2 has the largest posterior mean estimate of -6.69. The difference between these posterior mean estimates is very limited in comparison with the posterior mean estimates of the Bayesian lassos. Hence, the regularized horseshoe models show similar shrinkage behaviour on the large rbc_normal coefficient. Again, the small differences in posterior mean estimates are explainable by the MCMC sampling earlier explained.

The global shrinkage hyperparameter of RegHor2 was set on 0.1 which results in a prior distribution with a high peak and thin tails. The local parameter of RegHor3 was set on 0.1 which ensures heavy shrinkage on large coefficients. Based on this, we would expect heavy shrinkage on both su_1 and rbc_normal from RegHor2 and the heaviest shrinkage on rbc_normal from RegHor3. Figure 6 does show heavy shrinkage on the large coefficient with RegHor3, but the overall shrinkage behaviour of these models differ less than expected.

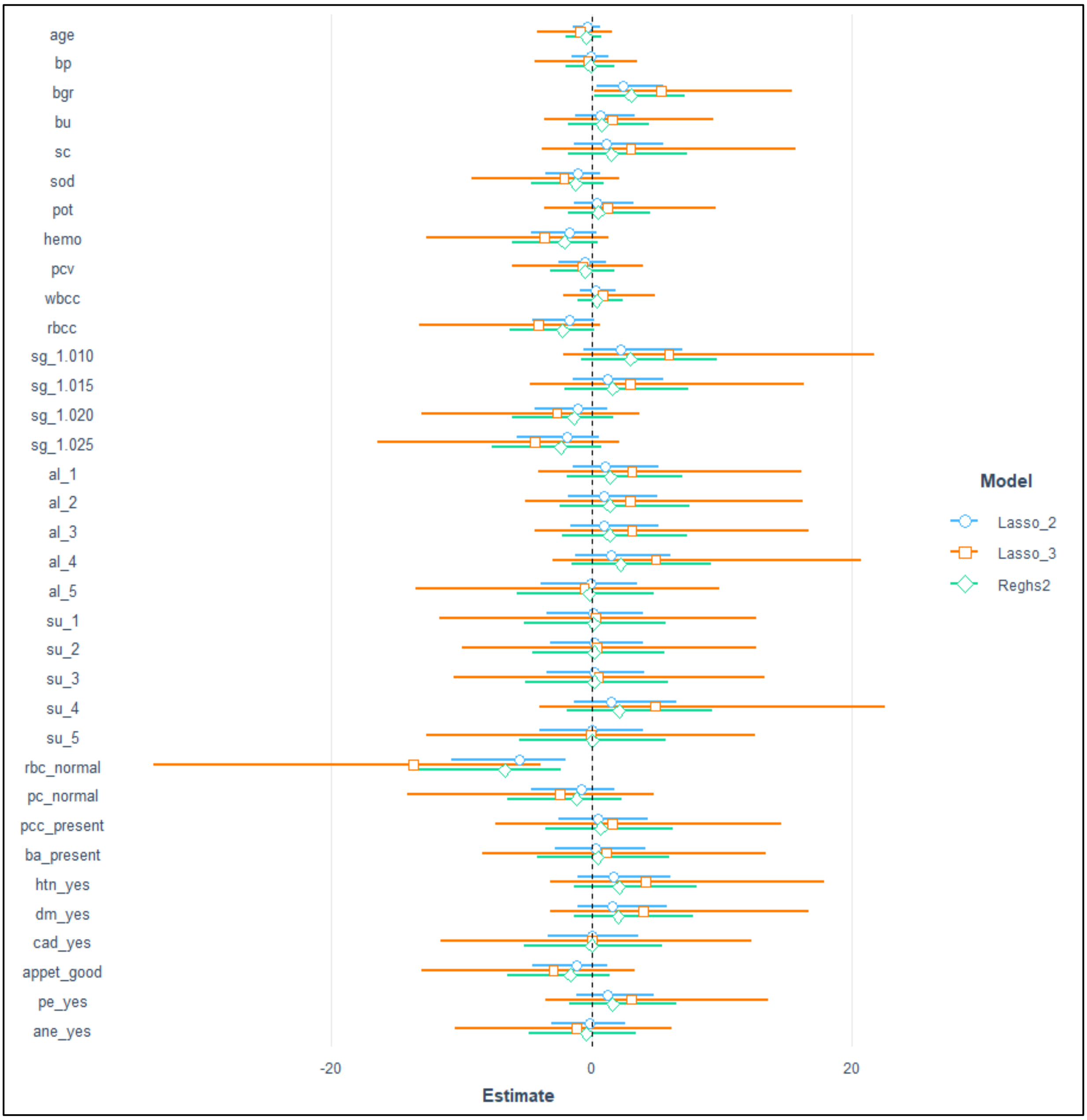

Figure 7 shows the comparison of posterior mean estimates and 95% credibility intervals.

In figure 7, the posterior mean estimates of all coefficients are given for lasso 2, lasso 3, and RegHor 2. These three models are compared because they all have a modified global shrinkage parameter, wherein lasso 2 and RegHor 2 have a low scale for heavy shrinkage, while lasso 3 has a higher df to ensure less shrinkage. The plot shows that lasso 2 and RegHor 2 shrink coefficients towards zero by a very comparable amount, wherein the shrinkage of lasso 2 is slightly heavier. This is the case for both small and large coefficients. Despite the comparable posterior mean estimates, we do see that RegHor2 has larger intervals than Lasso2. Larger intervals occur when the prior distribution is more spread out. Hence, the estimates of RegHor2 have a slightly higher uncertainty in the posterior mean estimates. Furthermore, we see that the least shrinkage is executed by lasso 3. The posterior mean estimates are the furthest from zero which implies less shrinkage. In addition, the intervals of lasso 3 are very large, but figure 2 also showed a very spread out lasso 3 prior. Hence, the interval and shrinkage behavior is in line with the expectation.

Variable selection

The cross-validated variable selection (varsel) function in `projpredict` is used to select variables based on their capability to lower the root mean square error (RMSE) of the model. There are multiple methods to obtain the selected variables with this function, the two used methods are RMSE plot interpretation and the suggest_size function. Variable selection with the RMSE plot is shown below.

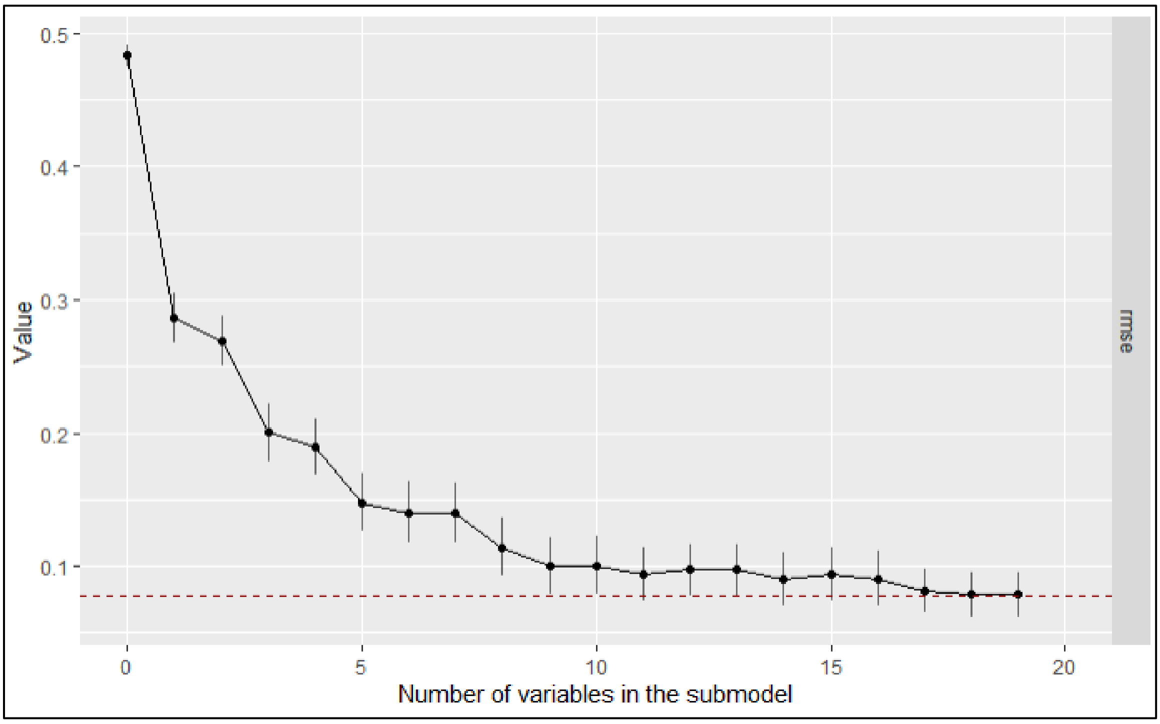

Figure 8 shows the RMSE plot of the Bayesian lasso 1 using projdpredict.

Figure 8 shows a decrease of the RMSE when the model size increases. Including more variables results in a lower RMSE, but could result in overfitting due to a high variance. The bias-variance trade-off should be balanced. To obtain this balance and model parsimony, only variables with a significant impact on the RMSE should be included. Figure 8 shows that after 9 variables, the decrease of RMSE is insignificant per extra included variable. Hence, the number of selected variables should be 9, based on the decrease of RMSE and the aim to realize model parsimony.

The second method to obtain the optimal model size is the suggest_size function. This function suggests a suitable model size based on certain model default settings. The decision guidelines for this function are heuristic, which implies that the outcome of this method should be considered as guidance. The number of selected variables for the lasso models are presented in in tables 6 and 7. The number of selected variables for the regularized horseshoe models are presented in in tables 8 and 9. All tables include the smallest and largest possible model as reference.

Table 6

The number of selected variables for Bayesian lasso based on RMSE plots

Model	Degrees of freedom	Scale	Selected variables	RMSE
Lasso 1 (default)	1	1	9	0.397
Lasso 2	1	0.1	9	0.397
Lasso 3	5	1	9	0.397
Smallest model	-	-	1	0.543
Largest model	-	-	35	0.401

Table 7

The number of selected variables for Bayesian lasso based on the suggested size function

Model	Degrees of freedom	Scale	Selected variables	RMSE
Lasso 1 (default)	1	1	14	0.402
Lasso 2	1	0.1	18	0.412
Lasso 3	5	1	16	0.407
Smallest model	-	-	1	0.543
Largest model	-	-	35	0.401

Table 6 shows that all lasso models have the same number of selected variables based on the RMSE plots. Thus, there is no significant RMSE decrease after a model size of 9. Table 7 shows that the suggest_size function results in larger model sizes, with even twice as many variables suggested for lasso 2 in comparison to the plot-based variable selection. The RMSE plot in figure 8 showed a slight increase of the RMSE value after variable number 14 which corresponds to number of selected variables for lasso 1 in table 7. Thus, the suggest_size function suggests the number of selected variables when the lowest RMSE value is followed by an increased RMSE. When minimizing the RMSE is the only consideration, the suggested model size of 14 could be used. However, this model size includes 5 variables with minor impact on the RMSE which endanger model parsimony.

Table 8

The number of selected variables for regularized horseshoe based on RMSE plots

Model	Global Scale	Scale slab	Suggested size	RMSE
RegHor 1 (default)	1	1	9	0.397
RegHor 2	0.1	1	8	0.397
RegHor 3	1	0.1	9	0.398
Smallest model	-	-	1	0.543
Largest model	-	-	35	0.401

Table 9

The number of selected variables for regularized horseshoe based on the suggest size functions

Model	Global Scale	Scale slab	Suggested size	RMSE
RegHor 1 (default)	1	1	17	0.404
RegHor 2	0.1	1	17	0.402
RegHor 3	1	0.1	18	0.409
Smallest model	-	-	1	0.543
Largest model	-	-	35	0.401

Table 8 shows that the regularized horseshoe models have the same model size as the lasso models based on the RMSE plots. These plots show that all extra included variables after a model size of 9 do not result in a significant decrease of the RMSE. Once more, we see that the suggest size function gives a higher number of selected variables for the models.

Model evaluation

To evaluate the predictive performances of the Bayesian lasso and regularized horseshoe models, the RMSE of the models are calculated and shown in table 6 to table 9. To get a better understanding of the model performances in terms of the RMSE, RMSE values of the smallest and largest possible models are also calculated and should be interpreted as upper and lower limits.

As shown earlier RMSE decreases when the model size increases. Thus, including only one variable in the model gives a high RMSE value of 0.543 and results in an underfitting model. Including all available variables of the dataset gives a lower RMSE value of 0.401 and results in an overfitting model. Surprisingly even though all models have a lower RMSE value than the upper limit, tables 6 and 8 show that models with a model size of 9 have a lower RMSE than the lower limit of 0.401.

Tables 6 to 9 also show that the RMSE of the Bayesian lasso and regularized horseshoe models are identical or close to each other. Both small and large Bayesian lasso and regularized horseshoe models have an RMSE close to the lower limit which suggests a good performing model. Hence, the different priors and hyperparameter settings did not result in different model performances in terms of RMSE.

Discussion

In this research, a comparison is performed between Bayesian penalized regression priors: lasso and regularized horseshoe. This study aimed to provide researchers with insights into the use of these priors to deal with high-dimensional data. Therefore, the shrinkage behaviour of the Bayesian lasso and regularized horseshoe models using different hyperparameter settings were compared. In addition, the lasso and regularized horseshoe models were evaluated based on their root-mean-square error (RMSE).

First, the global hyperparameter of the Bayesian lasso and regularized horseshoe were modified by lowering the scale to ensure heavy shrinkage on all coefficients. The heaviest shrinkage for both small and large coefficients was accomplished with the Bayesian lasso. However, across all specifications, this shrinkage did not differ much from the regularized horseshoe shrinkage. This result was in line with the expectation because the lower scaled lasso clearly showed a very peaked prior distribution with thin tails which results in heavy shrinkage for all coefficients. Furthermore, the different Bayesian lasso models showed more variation in shrinkage behavior than the different regularized horseshoe models. Literature shows that the regularized horseshoe prior is a robust prior which implies that the prior distribution is less dependent on specific hyperparameter settings (Piironen & Vehtari, 2017). Hence, both priors are suitable for heavy shrinkage of coefficients towards zero, but it is more complex to control the amount of shrinkage manually with the regularized horseshoe due to its robust property.

Second, variable selection on the models was performed using the RMSE plot and the suggest_size function which are available in the `projpred` package (Piironen et al., 2018). All Bayesian lasso and regularized horseshoe models had a model size of 9 based on their RMSE plot. This outcome is surprising for the lasso models because these models showed quite a few differences in shrinkage behavior on the coefficients. Hence, this method is not very sensitive for hyperparameter settings which is convenient because a balanced bias-variance trade-off can still be obtained when the chosen prior hyperparameter settings are not ideal. Furthermore, variable selection with the suggest_size function resulted in larger models in comparison to the RMSE plots. The decision rules of this function are heuristic and should be interpreted as guidance. The suggest_size function is not desirable because it includes more variables than necessary which could result in overfitting.

Third, the RMSE of the models were calculated to evaluate the model performances. Interesting is that the RMSE did not differ much between the small and large models. In that case, it is better to use the smaller model to achieve model parsimony and to ensure a balanced bias-variance trade-off. The Bayesian lasso and regularized horseshoe models with a model size of 9 had the lowest RMSE values. Thus, models with 9 selected variables perform better than the larger models for both priors. Despite the desirable low RMSE values, results also showed that both lasso and regularized horseshoe models had an RMSE value lower than the lower limit which should not occur. A possible explanation of the varying RMSE values is that the data was divided in a train and a test sets without a validation set. Literature shows that validation sets are used to provide an unbiased evaluation of a model on the training dataset while tuning model parameters (Bylander & Tate, 2006). Splitting the dataset only into train and test sets possibly resulted in biased or optimistic estimates. A solution to this problem is to implement the k-fold validation technique to deal with the bias (Rodríguez, Pérez, & Lozano, 2010). However, splitting the data into only a train and test set is a simpler and especially a faster way of cross-validation which is necessary for the slow Bayesian models, while k-fold validation requires more computation time.

This paper has several limitations. First, this research only compared the Bayesian lasso and regularized horseshoe priors, while Bayesian penalized regression includes more shrinkage priors such as the hyperlasso and the spike-and-slab. Second, the sample size of the data was large relative to the number of variables, while datasets with a small sample size relative to the number of variables cause most difficulties. Future research could investigate whether the Bayesian lasso and regularized horseshoe priors also show similar model performances in terms of RMSE at other settings such as a dataset with far more variables than observations.

Conclusion

Based on the findings of the comparison between the Bayesian lasso and the regularized horseshoe models, researchers should consider which prior suits their data best. The main consideration relates to the large coefficients of the data. The Bayesian lasso is a global prior, which has only two hyperparameters in `brms`: scale and degrees of freedom. Lowering the scale hyperparameter results in a very peaked distribution with thin tails which executes heavy shrinkage towards zero on all coefficients. Modifying the degrees of freedom to a larger value results in a less peaked distribution with thicker tails which results in less shrinkage. This prior does not allow large coefficients to escape from heavy shrinkage. The Bayesian regularized horseshoe on the other hand is a global-local prior and has five hyperparameters in `brms`: df, scale_global, df_global, scale_slab, and df_slab. Lowering the global scale did not result in more shrinkage than the default settings. In addition, lowering the local scale resulted in heavier shrinkage on the large coefficient but not significantly heavier than the default settings. Thus, the regularized horseshoe is a complex and robust prior which makes it difficult to easily interpret its hyperparameters, but generally the results seem robust to their specific settings. A second consideration is to choose the variable selection method. This research showed that the RMSE plot which is available in `projpredict` is very suitable for variable selection. This paper did not find significant differences in model predictive performances between the Bayesian lasso and regularized horseshoe.

Bibliography

Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., & Escobar, G. (2014). Big data in health care: Using analytics to identify and manage high-risk and high-cost patients. *Health Affairs*, *33*(7), 1123–1131. https://doi.org/10.1377/hlthaff.2014.0041

Bürkner, P. C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Bylander, T., & Tate, L. (2006). Using validation sets to avoid overfilling in adaBoost. In *FLAIRS 2006 - Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference* (Vol. 2006, pp. 544–549). Retrieved from www.aaai.org

Carvalho, C. M., Polson, N. G., & Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, *97*(2), 465–480. https://doi.org/10.1093/biomet/asq017

Derksen, S., & Keselman, H. J. (1992). Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, *45*(2), 265–282. https://doi.org/10.1111/j.2044-8317.1992.tb00992.x

Hans, C. (2009). Bayesian lasso regression. *Biometrika*, *96*(4), 835–845. https://doi.org/10.1093/biomet/asp047

Hawkins, D. M. (2004, January). The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*. American Chemical Society. https://doi.org/10.1021/ci0342472

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. *Applied Predictive Modeling*. https://doi.org/10.1007/978-1-4614-6849-3

Kyung, M., Gilly, J., Ghoshz, M., & Casellax, G. (2010). Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis*, *5*(2), 369–412. https://doi.org/10.1214/10-BA607

Li, Q., & Liny, N. (2010). The Bayesian elastic net. *Bayesian Analysis*, *5*(1), 151–170. https://doi.org/10.1214/10-BA506

McNeish, D. M. (2015). Using Lasso for Predictor Selection and to Assuage Overfitting: A Method Long Overlooked in Behavioral Sciences. *Multivariate Behavioral Research*, *50*(5), 471–484. https://doi.org/10.1080/00273171.2015.1036965

Park, T., & Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association*, *103*(482), 681–686. https://doi.org/10.1198/016214508000000337

Piironen, J., Paasiniemi, M., & Vehtari, A. (2018). projpred: Projection Predictive Feature Selection. Retrieved from https://mc-stan.org/projpred/

Piironen, J., & Vehtari, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, *11*(2), 5018–5051. https://doi.org/10.1214/17-EJS1337SI

Polson, N. G., & Scott, J. G. (2010). Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction. In *Bayesian Statistics 9* (Vol. 9780199694). https://doi.org/10.1093/acprof:oso/9780199694587.003.0017

Rodríguez, J. D., Pérez, A., & Lozano, J. A. (2010). Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(3), 569–575. https://doi.org/10.1109/TPAMI.2009.187

Tibshiranit, R. (1996). *Regression Shrinkage and Selection via the Lasso*. *J. R. Statist. Soc. B* (Vol. 58).

UCI. (2015). UCI Machine Learning Repository: Chronic_Kidney_Disease Data Set. Retrieved July 10, 2021, from https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease#

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). *mice: Multivariate imputation by chained equations in R*. *Journal of Statistical Software* (Vol. 45). https://doi.org/10.18637/jss.v045.i03

van Erp, S. (2020). A Tutorial on Bayesian Penalized Regression with Shrinkage Priors for Small Sample Sizes. *Small Sample Size Solutions*, 71–84. https://doi.org/10.4324/9780429273872-6

van Erp, S., Oberski, D. L., & Mulder, J. (2019, April 1). Shrinkage priors for Bayesian penalized regression. *Journal of Mathematical Psychology*. Academic Press Inc. https://doi.org/10.1016/j.jmp.2018.12.004

Appendix

I. RMSE plots

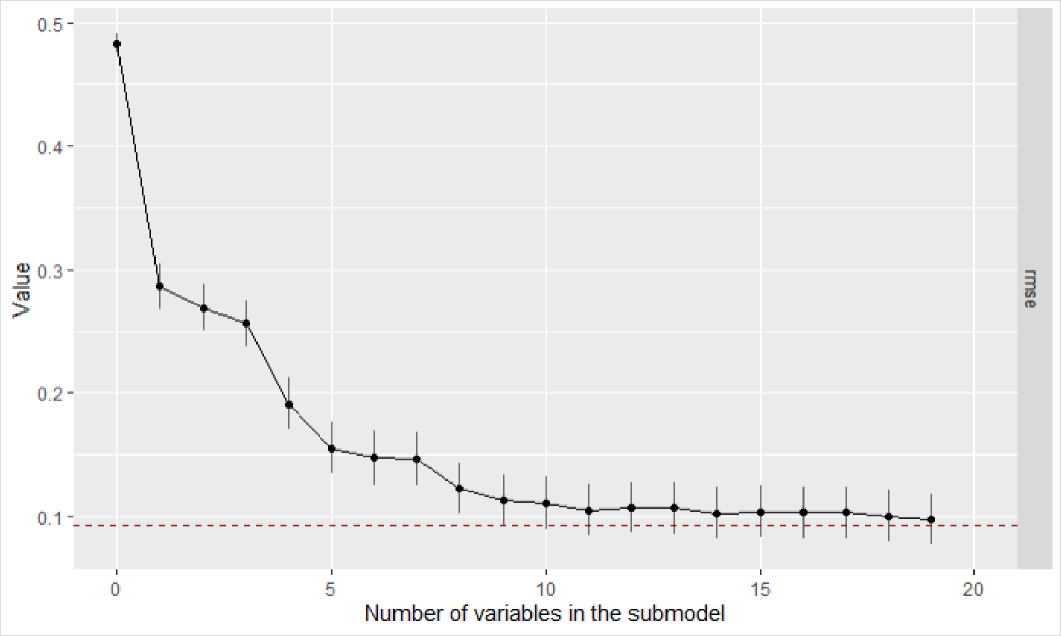

Figure 9 Projpred RMSE plot of Lasso 2.

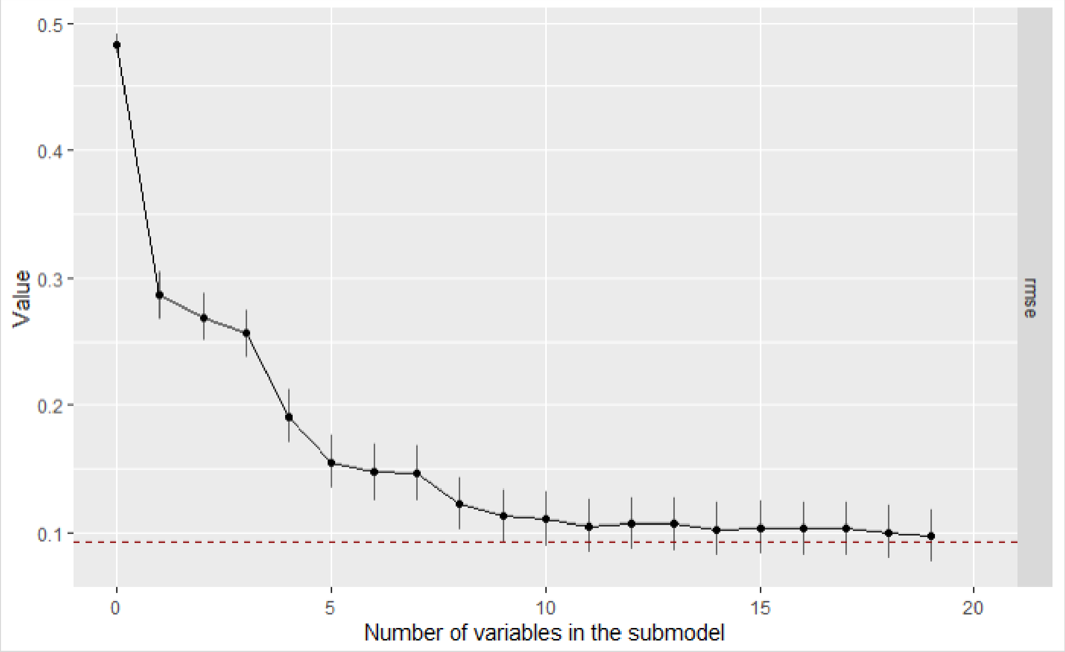

Figure 10 Projpred RMSE plot of Lasso 3

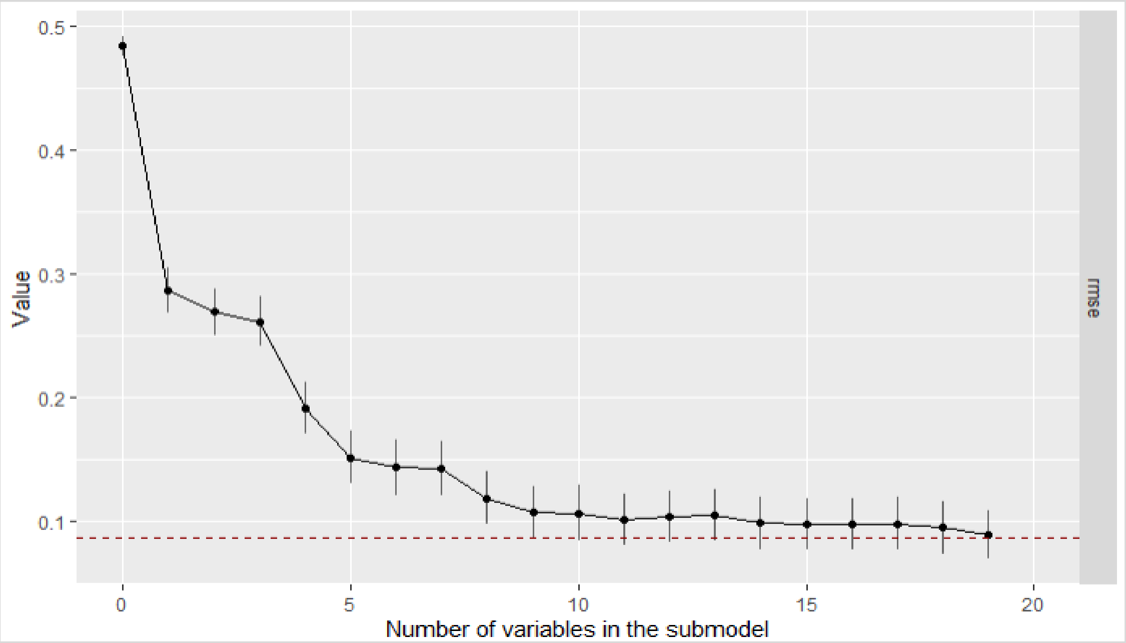

Figure 11: Projpred RMSE plot of regularized horseshoe 1.

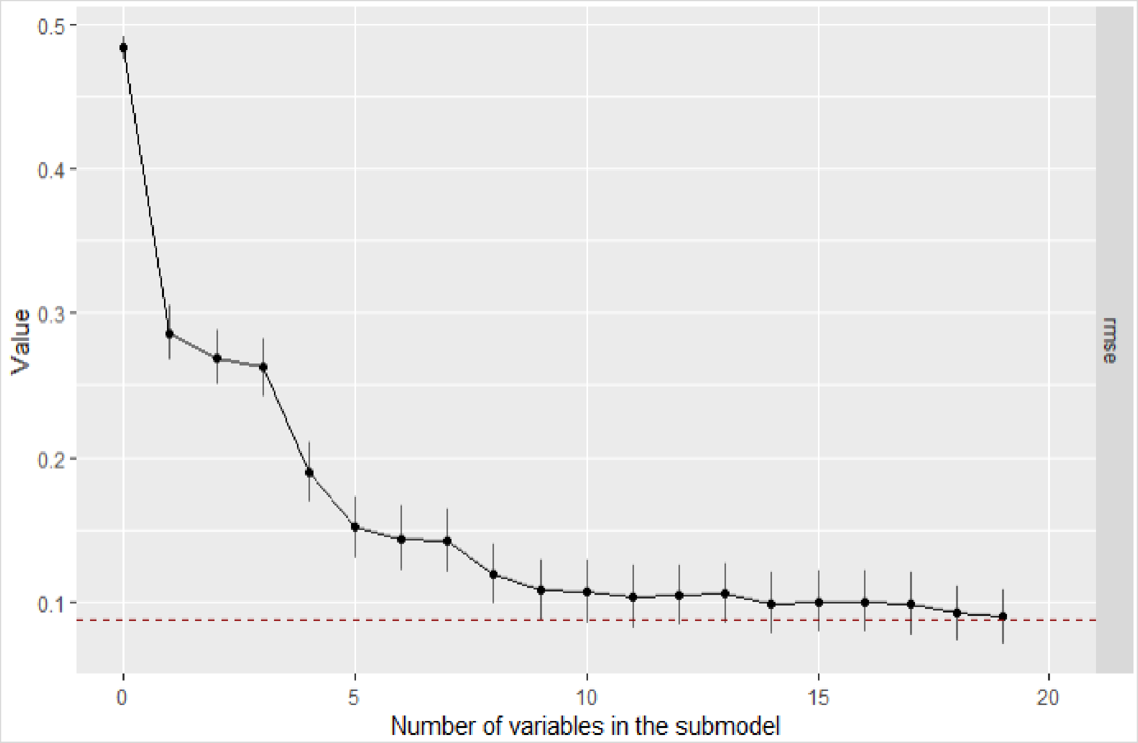

Figure 12: Projpred RMSE plot of regularized horseshoe 2.

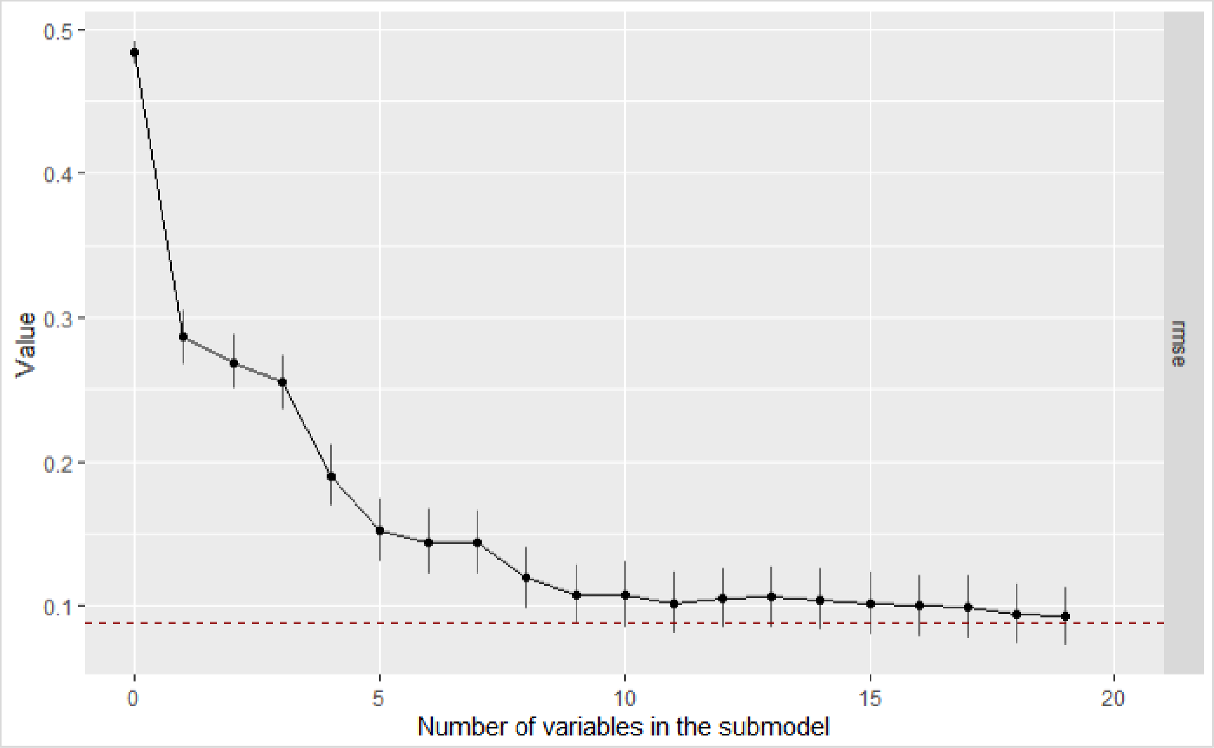

Figure 13: : Projpred RMSE plot of regularized horseshoe 3.

II. R packages

```
##  R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Dutch_Netherlands.1252  LC_CTYPE=Dutch_Netherlands.1252
## [3] LC_MONETARY=Dutch_Netherlands.1252 LC_NUMERIC=C
## [5] LC_TIME=Dutch_Netherlands.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] LaplacesDemon_16.1.4 invgamma_1.1         extraDistr_1.9.1
##  [4] Metrics_0.1.4        projpred_2.0.2       rstanarm_2.21.1
##  [7] bayesplot_1.8.0      broom.mixed_0.2.6    ggstance_0.3.5
## [10] jtools_2.1.3         rstantools_2.1.1     brms_2.15.0
## [13] Rcpp_1.0.6           reshape2_1.4.4       ggplot2_3.3.3
## [16] fastDummies_1.6.3    mice_3.13.0          dplyr_1.0.6
## [19] RWeka_0.4-43
##
## loaded via a namespace (and not attached):
##   [1] minqa_1.2.4          colorspace_2.0-1     ellipsis_0.3.2
##   [4] ggridges_0.5.3       rsconnect_0.8.18     estimability_1.3
##   [7] markdown_1.1         base64enc_0.1-3      rstan_2.21.2
##  [10] DT_0.18              fansi_0.5.0          mvtnorm_1.1-1
##  [13] bridgesampling_1.1-2 codetools_0.2-18     splines_4.1.0
##  [16] knitr_1.33           shinythemes_1.2.0    jsonlite_1.7.2
##  [19] nloptr_1.2.2.2       rJava_1.0-4          broom_0.7.6
##  [22] shiny_1.6.0          compiler_4.1.0       emmeans_1.6.1
##  [25] backports_1.2.1      assertthat_0.2.1     Matrix_1.3-3
##  [28] fastmap_1.1.0        cli_2.5.0            later_1.2.0
##  [31] htmltools_0.5.1.1    prettyunits_1.1.1    tools_4.1.0
##  [34] igraph_1.2.6         coda_0.19-4          gtable_0.3.0
##  [37] glue_1.4.2           V8_3.4.2             vctrs_0.3.8
##  [40] nlme_3.1-152         crosstalk_1.1.1      xfun_0.23
##  [43] stringr_1.4.0        ps_1.6.0             lme4_1.1-27
##  [46] mime_0.10            miniUI_0.1.1.1       lifecycle_1.0.0
##  [49] gtools_3.8.2         RWekajars_3.9.3-2    MASS_7.3-54
##  [52] zoo_1.8-9            scales_1.1.1         colourpicker_1.1.0
##  [55] promises_1.2.0.1     Brobdingnag_1.2-6    parallel_4.1.0
##  [58] inline_0.3.19        TMB_1.7.20           shinystan_2.5.0
##  [61] gamm4_0.2-6          yaml_2.2.1           curl_4.3.1
##  [64] gridExtra_2.3        pander_0.6.3         loo_2.4.1
##  [67] StanHeaders_2.21.0-7 stringi_1.6.1        dygraphs_1.1.1.6
##  [70] boot_1.3-28          pkgbuild_1.2.0       rlang_0.4.11
##  [73] pkgconfig_2.0.3      matrixStats_0.58.0   evaluate_0.14
##  [76] lattice_0.20-44      purrr_0.3.4          htmlwidgets_1.5.3
##  [79] tidyselect_1.1.1     processx_3.5.2       plyr_1.8.6
##  [82] magrittr_2.0.1       R6_2.5.0             generics_0.1.0
##  [85] DBI_1.1.1            pillar_1.6.1         withr_2.4.2
##  [88] mgcv_1.8-35          xts_0.12.1           survival_3.2-11
##  [91] abind_1.4-5          tibble_3.1.2         crayon_1.4.1
##  [94] utf8_1.2.1           rmarkdown_2.8        grid_4.1.0
##  [97] callr_3.7.0          threejs_0.3.3        digest_0.6.27
## [100] xtable_1.8-4         tidyr_1.1.3          httpuv_1.6.1
## [103] RcppParallel_5.1.4   stats4_4.1.0         munsell_0.5.0
## [106] shinyjs_2.0.0
```